In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [5]:

```
df = pd.read_csv(r"C:\Users\Dell\Downloads\Iris.csv.")
```

In [6]:

```
df
```

Out[6]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

In [7]:

```
df.head()
```

Out[7]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

In [8]:

```
df.tail()
```

Out[8]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

In [9]:

```
df.columns
```

Out[9]:

```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthC
m',
       'Species'],
      dtype='object')
```

In [10]:

```
df.describe()
```

Out[10]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [11]:

```
df.info
```

Out[11]:

```
<bound method DataFrame.info of       Id  SepalLengthCm  SepalWidthCm  Petal
LengthCm  PetalWidthCm  \
0      1            5.1           3.5            1.4           0.2
1      2            4.9           3.0            1.4           0.2
2      3            4.7           3.2            1.3           0.2
3      4            4.6           3.1            1.5           0.2
4      5            5.0           3.6            1.4           0.2
..   ...            ...           ...            ...           ...
145  146            6.7           3.0            5.2           2.3
146  147            6.3           2.5            5.0           1.9
147  148            6.5           3.0            5.2           2.0
148  149            6.2           3.4            5.4           2.3
149  150            5.9           3.0            5.1           1.8

            Species
0       Iris-setosa
1       Iris-setosa
2       Iris-setosa
3       Iris-setosa
4       Iris-setosa
..              ...
145  Iris-virginica
146  Iris-virginica
147  Iris-virginica
148  Iris-virginica
149  Iris-virginica

[150 rows x 6 columns]>
```

In [12]:

```
from sklearn.preprocessing import LabelEncoder
```

In [13]:

```
LE = LabelEncoder()
```

In [14]:

```
df.iloc[:,-1] = LE.fit_transform(df.iloc[:,-1])
```

In [15]:

```
df
```

Out[15]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	0
1	2	4.9	3.0	1.4	0.2	0
2	3	4.7	3.2	1.3	0.2	0
3	4	4.6	3.1	1.5	0.2	0
4	5	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	2
146	147	6.3	2.5	5.0	1.9	2
147	148	6.5	3.0	5.2	2.0	2
148	149	6.2	3.4	5.4	2.3	2
149	150	5.9	3.0	5.1	1.8	2

150 rows × 6 columns

In [17]:

```
df = df.sample(frac=1)
```

In [18]:

```
df.head(10)
```

Out[18]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
68	69	6.2	2.2	4.5	1.5	1
139	140	6.9	3.1	5.4	2.1	2
69	70	5.6	2.5	3.9	1.1	1
26	27	5.0	3.4	1.6	0.4	0
55	56	5.7	2.8	4.5	1.3	1
38	39	4.4	3.0	1.3	0.2	0
67	68	5.8	2.7	4.1	1.0	1
77	78	6.7	3.0	5.0	1.7	1
25	26	5.0	3.0	1.6	0.2	0
124	125	6.7	3.3	5.7	2.1	2

In [22]:

```
x = df.iloc[:,:-1]
x.head()
```

Out[22]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
68	69	6.2	2.2	4.5	1.5
139	140	6.9	3.1	5.4	2.1
69	70	5.6	2.5	3.9	1.1
26	27	5.0	3.4	1.6	0.4
55	56	5.7	2.8	4.5	1.3

In [23]:

```
y =df.iloc[:,-1]
y.head()
```

Out[23]:

```
68     1
139    2
69     1
26     0
55     1
Name: Species, dtype: int32
```

In [24]:

```
from sklearn.model_selection import train_test_split
```

In [25]:

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_state=50)
```

In [26]:

```
X_train.head()
```

Out[26]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
12	13	4.8	3.0	1.4	0.1
116	117	6.5	3.0	5.5	1.8
130	131	7.4	2.8	6.1	1.9
60	61	5.0	2.0	3.5	1.0
22	23	4.6	3.6	1.0	0.2

In [27]:

```
X_train.shape
```

Out[27]:

```
(120, 5)
```

In [28]:

```
from sklearn.tree import DecisionTreeClassifier
```

In [29]:

```
dt = DecisionTreeClassifier()
```

In [30]:

```
dt.fit(X_train,y_train)
```

Out[30]:

```
DecisionTreeClassifier()
```

In [31]:

```
from sklearn.metrics import classification_report,confusion_matrix
```

In [32]:

```
y_predicted = dt.predict(X_test)
```

In [33]:

```
y_predicted
```

Out[33]:

```
array([0, 1, 1, 0, 0, 1, 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 2, 2, 2, 1, 0, 2,
       2, 2, 0, 2, 1, 1, 2, 1])
```

In [34]:

```
y_test = np.array(y_test)
```

In [35]:

```
y_test
```

Out[35]:

```
array([0, 1, 1, 0, 0, 1, 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 2, 2, 2, 1, 0, 2,
       2, 2, 0, 2, 1, 1, 2, 1])
```

In [36]:

```
from sklearn.metrics import accuracy_score
```

In [37]:

```
accuracy_score(y_predicted, y_test)
```

Out[37]:

```
1.0
```

In [38]:

```
from sklearn.metrics import confusion_matrix
```

In [39]:

```
confusion_matrix(y_predicted, y_test)
```

Out[39]:

```
array([[ 8,  0,  0],
       [ 0,  7,  0],
       [ 0,  0, 15]], dtype=int64)
```

In [40]:

```
from sklearn .metrics import classification_report
```

In [41]:

```
print(classification_report(y_predicted, y_test))
```

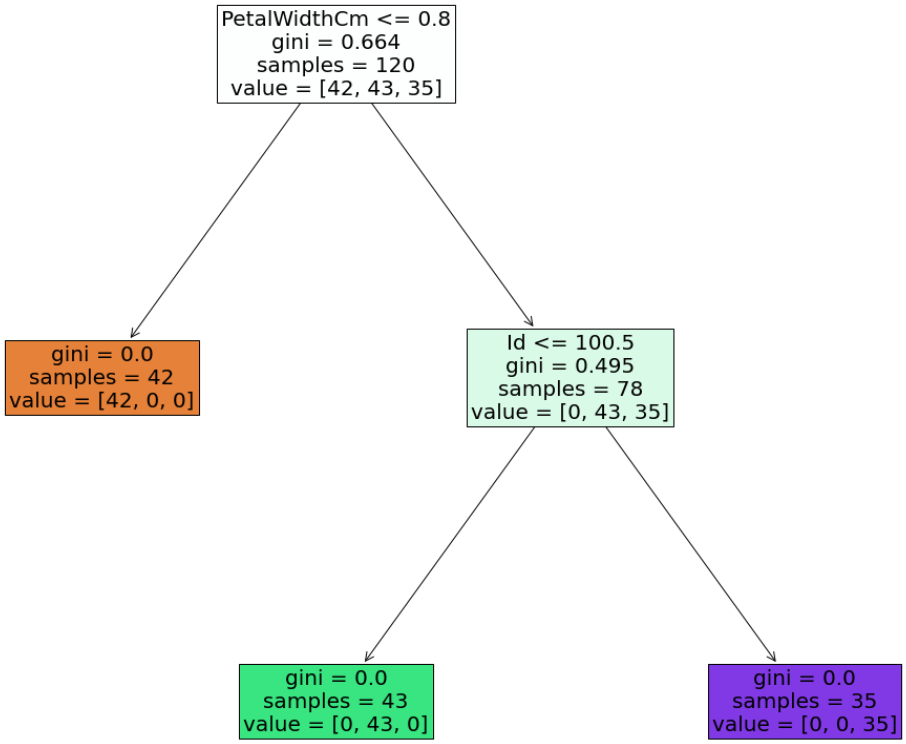
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         8
           1       1.00      1.00      1.00         7
           2       1.00      1.00      1.00        15

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

In [42]:

```
from sklearn import tree
```

In [43]:

```
plt.figure(figsize=(20,17))
dtviz = tree.plot_tree(dt,feature_names = x.columns, filled = True, fontsize =20)
```



In [44]:

```
# THANK YOU
```

In [ ]:

```

```